

NOV 15 2004

PROVISIONAL APPLICATION FOR PATENT COVER SHEET (Large Entity)

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(S)/APPLICANT(S)

Given Name (first and middle (if any))	Family Name or Surname	Residence (City and either State or Foreign Country)
Neil Kevin Scott	Enns Kennedy Shell	

☐ Additional inventors are being named on page 2 attached hereto**TITLE OF THE INVENTION (280 characters max)**

FLEXIBLE DISPLAY OF INFORMATION ON MOBILE DEVICES

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☒ Customer Number 022913

OR

☒ Firm or Individual Name WORKMAN, NYDEGGER & SEELEY, C/O RICK D. NYDEGGER. PATENT TRADEMARK OFFICE

Address 1000 EAGLE GATE TOWER

Address 60 EAST SOUTH TEMPLE

City SALT LAKE CITY State UTAH ZIP 84111

Country USA Telephone 801-533-9800 Fax 801-328-1707

ENCLOSED APPLICATION PARTS (check all that apply)☒ Specification Number of Pages 4☐ Drawing(s) Number of Sheets☒ Other (specify) FORM PTO-2038**METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)**☒ A check or money order is enclosed to cover the filing fees☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number:

23-3178

FILING FEE
AMOUNT (\$)

\$150.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government

☒ No☐ Yes, the name of the U S Government agency and the Government contract number are

Respectfully submitted,

SIGNATURE

DATE

OCTOBER 2, 2000

TYPED or PRINTED NAME ADRIAN J. LEE

REGISTRATION NO.
(if appropriate)

42,785

TELEPHONE

801-533-9800

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, DC 20231

CERTIFICATE OF MAILING BY "EXPRESS MAIL" (37 CFR 1.10)

Applicant(s): Neil Enns and Kevin Kennedy

Docket No.

13768.183

Serial No.

Not Yet Assigned

Filing Date

October 2, 2000

Examiner

Not Yet Assigned

Group Art Unit

Not Yet Assigned

Invention: FLEXIBLE DISPLAY OF INFORMATION ON MOBILE DEVICES

RECEIVED

NOV 2 2 2004

Technology Center 2600

I hereby certify that this correspondence listed as transmitted below

(Identify type of correspondence)

is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under

37 CFR 1.10 in an envelope addressed to: The Assistant Commissioner for Patents, Washington, D.C. 20231 on

October 2, 2000

(Date)

Lisa L. Rogers

(Typed or Printed Name of Person Mailing Correspondence)

(Signature of Person Mailing Correspondence)

EL 585 552 875 US

("Express Mail" Mailing Label Number)

EL585552875US

Note: Each paper must have its own certificate of mailing.

Transmitted:

- ▶ Provisional Application 4 pgs
- ▶ Patent Cover Sheet 1 pg.
- ▶ Form PTO-2038 (\$150.) 1 pg.
- ▶ Certificate of Mailing 1 pg.
- ▶ Postcard

160462.1 – Home Screen

Title of Invention: Flexible display of information on mobile devices

Document Author(s): Neil Enns and Kevin Kennedy

Introduction

Every mobile phone on the market has a 'home screen' where the user starts dialing. This screen typically displays the carrier name, battery and signal strength, and provides some mechanism for accessing other functionality on the phone.

We've taken this concept a step further by making the home screen fully customisable by the user. Layouts can be created using an XML-based authoring language that use plug-ins to display information on the page. This information can range from your current carrier name to the latest sports scores to stock quotes. It's very similar to the Digital Dashboard concept from the desktop.

Throughout this document, the following two terms are used heavily. I'm going to define them now in hopes that it will help make the rest of the document and the invention easier to understand.

Layout: A text document, written using an XML-based language, that defines the look of a home screen. It consists of sections that include plug-ins, and each section defines the position, fonts, colours, and display of the plug-ins. Advanced users, graphic designers, OEMs, Carriers, etc. can easily author their own layouts.

Plug-ins: Typically a .DLL that a software developer has written to work in our home screen framework. Plug-ins display information using the formatting information obtained from the layout. An example of a plug-in is one that displays the current carrier name. Another example is a plug-in that displays a 4-day weather forecast.

Motivation for the Invention:

When we were writing the specifications for our smartphone product, we spent a lot of time trying to come up with the one "right" main screen layout that would satisfy the needs of our users. It took us a long time, but we finally realised that there was no "one right" layout. Different people have different needs, and want to see different information when they glance at their phone. Further, a fixed layout does not provide carriers and OEMs with an opportunity to include branded content on the home screen.

The Pocket PC had previously done some work in this area with their Today page. Software developers could write plug-ins and display them on the Today screen of the device. However, this approach meant that only software developers could customise the look of the plug-ins (colours, font sizes, etc.). The average user could not put together a layout and easily change the visual appearance of the plug-ins to suit their tastes.

We solved these problems by creating a home screen with an XML-based language and plug-ins. Instead of having a single home screen, users can now have multiple layout files installed and can switch between the different layouts. To accomplish this we had to invent:

- 1) The XML-based language for defining the look of an layout
- 2) The APIs and methods that the plug-ins use to obtain layout information from the XML file
- 3) A notification scheme so plug-ins can find information out about the state of the device without using too much battery power

Description of the Invention:

Since there are several parts to this invention, to simplify things here are descriptions for each part:

The XML-Based Language

The language is fully documented in the Home Screen specification. It's a fairly simple markup language that can be extended on a per-plug-in basis if needed. Here is a sample of the XML that draws a clock using Arial 10pt in red, using 12-hour format:

```
<plugin file="sysplug.dll" name="clock">
  <time bgcolor="#FFFFFF" fgcolor="#FF0000" font-face="Arial"
font-size="10" mode="12"/>
</plugin>
```

Here is an example that draws the same clock using Times New Roman 12pt in blue, using a 24-hour format:

```
<plugin file="sysplug.dll" name="clock">
  <time bgcolor="#FFFFFF" fgcolor="#0000FF" font-face="Times New Roman"
font-size="12" mode="24"/>
</plugin>
```

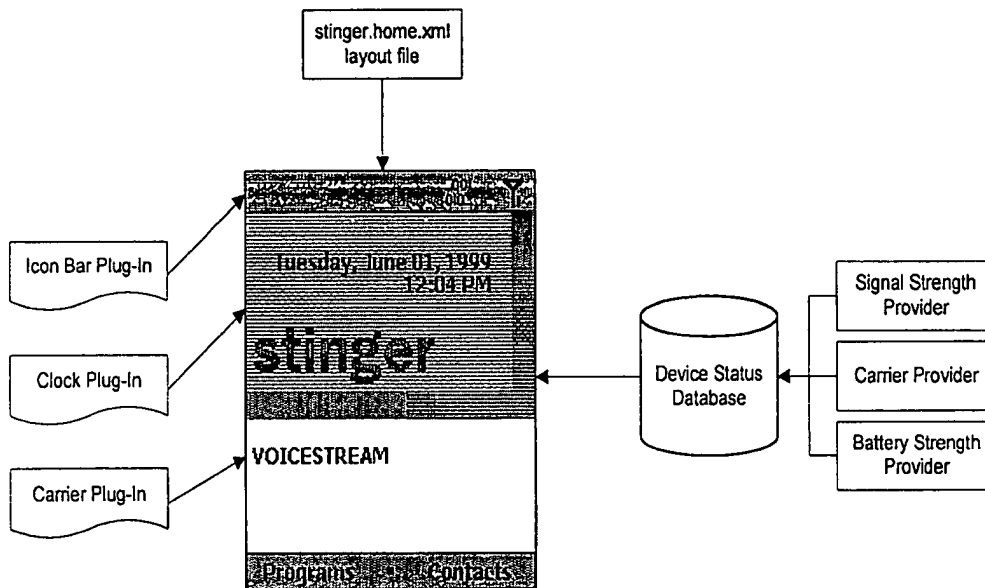
The Notification Mechanism

The need to display data from disparate applications in the system with high performance is the primary motivator for the current design. Rather than have the individual status producing applications be queried for the data directly by the plug-in, the data is placed by the status producing application in to an intermediate "status store." Status consuming applications that need to display this status in an up-to-date fashion will register themselves with the status store. When the data in the store changes, notifications go out to the registered applications. The registered applications are told which data changed. If they are interested in any of that data, the registered application queries the status store to get the most up to date version. Diagram 8 shows the flow of updates in the system from data producing applications to data consuming applications.

The home screen is a status consuming application. Each plug-in has specific status elements it is interested in. A "plug-in manager" keeps track of what data each plug-in wants updates for. When the data changes in the status store, the plug-in manager is notified. The plug-in manager runs through its list of plug-ins and causes the ones whose data has changed to redraw. During the redraw, the plug-in requests the updated data from the plug-in manager and displays it. The plug-in manager passes data requests directly to the status store. Diagram 8.1 has a UML sequence diagram of this.

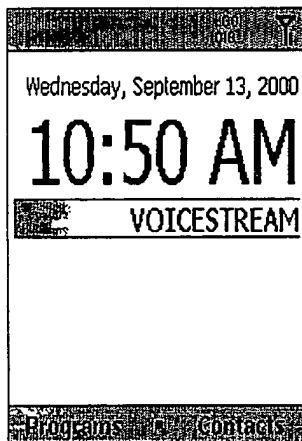
There are many advantages to this mechanism. Accessing data in the store is very fast compared to querying it directly from the owning application. The plug-in uses only a single interface to access all the data it needs to display. The plug-in does not have to cache data that is expensive to acquire, like the phone's signal strength. If data is not immediately available, say at boot when data producing applications are still initializing, the plug-in will be notified when the data is ready. The alternative is for the plug-in to be inactive while it is waiting.

Diagrams and Flow Charts:



In the above figure, the home screen reads the stinger.home.xml layout file. The file references three plug-ins that the home screen architecture loads. The plug-ins render their information based on the formatting information in the layout file. Status, such as the carrier name and signal strength, are updated in the device database by external data providers. When the data changes, the plug-ins are notified so they can update their display.

Here is a picture of another home screen that uses the same three plug-ins, but has different formatting options in the XML file:



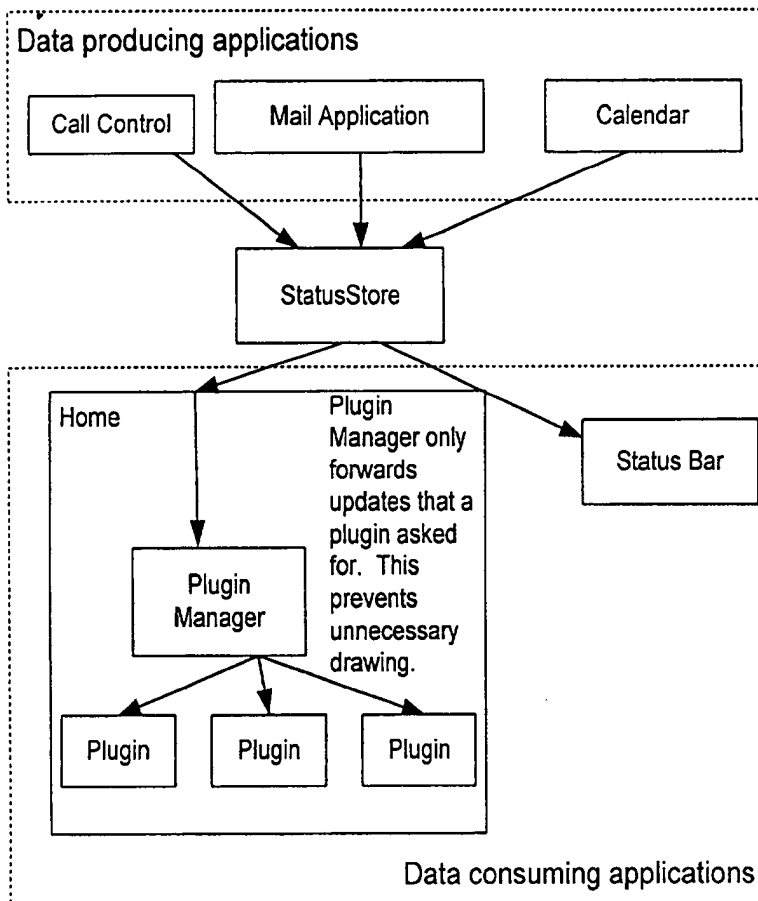


Diagram X - Flow of updates through system

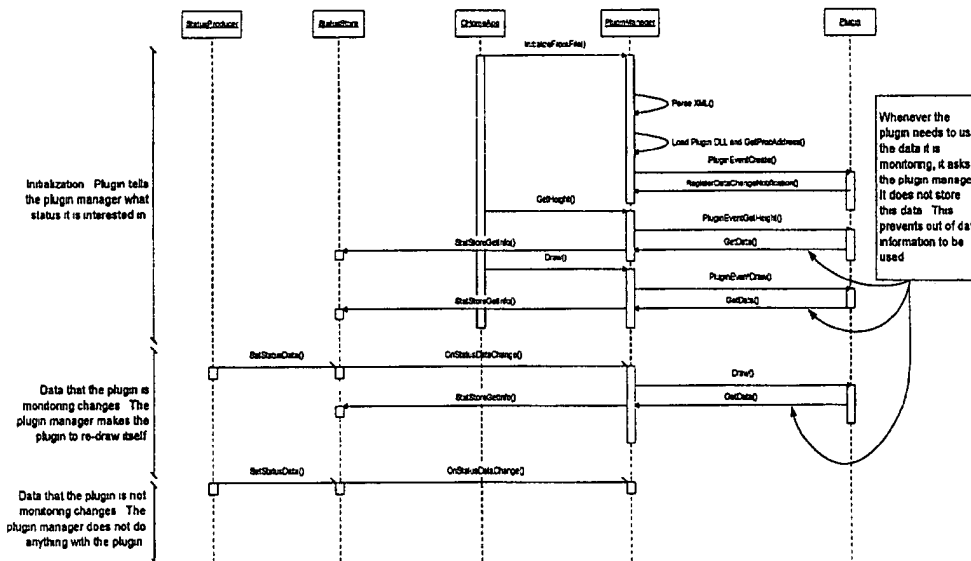


Diagram X+1 - UML sequence diagram of home screen initialization and status change notification handling.